

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

ALGORITMOS AVANZADOS

5ta. práctica (tipo B)
(Primer Semestre 2025)

Duración: 1h 50 min.

- **No puede utilizar apuntes, solo hojas sueltas en blanco.**
- En cada función el alumno deberá incluir, a modo de comentario, la forma de solución que utiliza para resolver el problema. De no incluirse dicho comentario, el alumno perderá el derecho a reclamo en esa pregunta.
- No puede emplear plantillas o funciones no vistas en los cursos de programación de la especialidad.
- Los programas deben ser desarrollados en el lenguaje C++. Si la implementación es diferente a la estrategia indicada o no la incluye, la pregunta no será corregida.
- Un programa que no muestre resultados coherentes y/o útiles será corregido sobre el 50% del puntaje asignado a dicha pregunta.
- Debe utilizar comentarios para explicar la lógica seguida en el programa elaborado. El orden será parte de la evaluación.
- Se utilizarán herramientas para la detección de plagios, por tal motivo si se encuentran soluciones similares, se anulará la evaluación a todos los implicados y se procederá con las medidas disciplinarias dispuestas por la FCI.
- **Solo está permitido acceder a la plataforma de PAIDEIA, cualquier tipo de navegación, búsqueda o uso de herramientas de comunicación se considera plagio por tal motivo se anulará la evaluación y se procederá con las medidas disciplinarias dispuestas por la FCI.**
- Para esta evaluación solo se permite el uso de las librerías **iostream, iomanip, climits, cmath, fstream, vector, map, iterator, algorithm o cstring**
- Su trabajo deberá ser subido a PAIDEIA.
- **Es obligatorio usar como compilador NetBeans.**
- Los archivos deben llevar como nombre su código de la siguiente forma `codigo_LAB5_P#` (donde # representa el número de la pregunta a resolver)

Pregunta 1 (20 puntos)

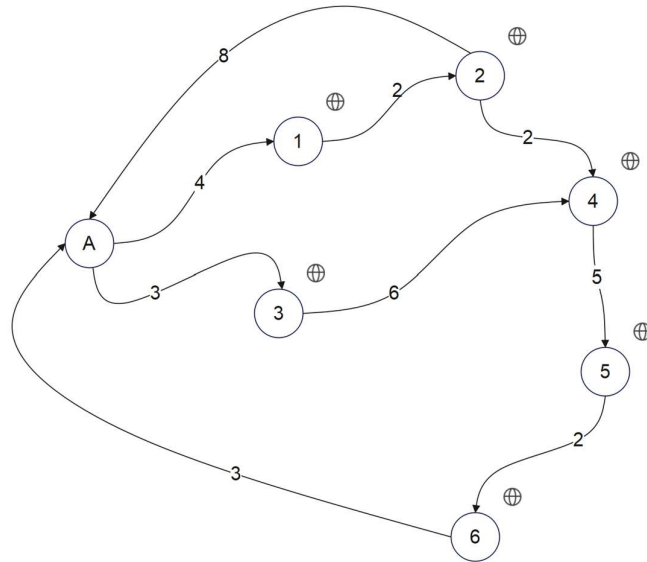
El problema del m-TSP consiste en lo siguiente: una empresa de reparto de productos vendidos de manera on-line tiene que visitar N clientes empleando K vehículos (que conoceremos como repartidores). Todos los repartidores salen y regresan del depósito central de productos de la empresa de reparto con sus respectivas guías de remisión que indican qué clientes deben visitar para entregarles sus productos.

El objetivo es: minimizar la distancia total recorrida por todos los repartidores, **cumpliendo además que cada cliente sea visitado exactamente una vez y por un solo repartidor.**

Nota: la solución del problema está constituida por un grupo de rutas, no una sola.

Ejemplo: en el siguiente grafo

- **Depósito:** Nodo A
- **Clientes:** hay que repartir productos en los nodos 1, 2, 3, 4, 5, 6
- **Repartidores:** K = 2



La respuesta estaría conformada por las siguientes rutas:

- **Repartidor 1:**
Ruta: A → Cliente 1 → Cliente 2 → A. Distancia 1 = 4+2+8 = 14
- **Repartidor 2:**
Ruta: A → Cliente 3 → Cliente 4 → Cliente 5 → Cliente 6 → A. Distancia 2 = 3+6+5+2+3 = 19

Todos los clientes han sido atendidos, por lo que es una respuesta válida con una distancia total = 33 unidades de longitud.

Se pide desarrollar un algoritmo genético que resuelva este problema considerando lo siguiente:

- a) Cromosoma de los individuos de la población
- b) Función de fitness
- c) Procedimientos detectores de aberraciones
- d) Ruleta con selección natural y probabilidades
- e) Operador de casamiento
- f) Operador de mutación o inversión

El programa deberá generar al menos 100 poblaciones y deberá devolver las rutas óptimas para los K repartidores, junto con la distancia total acumulada.

Consideraciones de diseño

1. Siendo m la cantidad de visitas que se tienen que hacer, el grafo puede tener más nodos
2. Al ser el inicio y el fin el mismo nodo (A) puede omitirse del cromosoma de todos los individuos.

CROMOSOMA

El cromosoma está formado por las rutas completas de los K repartidores. O sea, todos los cromosomas van a tener m genes. **Para su solución considere K=2.**

Ejemplos: para el Individuo I1 =(1,2,3,4,5,6) y para el individuo I2= (3,4,5,6,1,2)

Adicionalmente, a cada individuo se le tiene que asociar una lista denominada **cambio_de_repartidor**, que nos indica en qué gen cambiamos a otro vehículo repartidor R i-ésimo.

Ejemplos: para el Individuo I1= (1,2,3,4,5,6) la lista **cambio_de_repartidor** = {3} porque en el tercer gen comienza la ruta del segundo repartidor y para el individuo I2= (3,4,5,6,1,2) la lista **cambio_de_repartidor** = {5} porque en el quinto gen comienza la ruta del segundo repartidor.

Finalmente, el Individuo es una estructura formada por su cromosoma y su lista de cambio de repartidor.

FITNESS: La función estará dada por la distancia total de ambos repartidores y se busca minimizar esta distancia.

CLONES: Dos individuos son clones si cumplen simultáneamente:

- Tienen exactamente la misma secuencia de clientes (la misma permutación del cromosoma).
- Tienen exactamente la misma lista de cortes.

ABERRACIONES

TIPO DE ABERRACIÓN	CAUSA
A. CROMOSOMA NO PERMUTACIONAL	REPETICIÓN O AUSENCIA DE ALGÚN CLIENTE EN EL CROMOSOMA
B. RUTA INVÁLIDA SEGÚN EL GRAFO	ENTRE DOS CLIENTES CONSECUTIVOS (O ENTRE A Y CLIENTE, O CLIENTE Y A), NO EXISTE UN ARCO DIRECTO SEGÚN LA MATRIZ DE ADYACENCIA
C. VENDEDOR SIN CLIENTES (SI ESTÁ PROHIBIDO)	UNA SEGMENTACIÓN QUE GENERA ALGUNA RUTA VACÍA
D. CICLOS DENTRO DE LA RUTA (SOLO PARA CIERTAS VARIANTES)	QUE UN VENDEDOR RECORRA EL MISMO NODO CLIENTE DOS VECES

Al finalizar el laboratorio, comprima la carpeta de su proyecto empleando el programa Zip que viene por defecto en el Windows, **no se aceptarán los trabajos compactados con otros programas como RAR, WinRAR, 7zip o similares**. Luego súbalo a la tarea programa en Paideia para este examen.

Profesores del curso:

Manuel Tupia
Rony Cueva
Igor Siveroni

San Miguel, 28 de junio del 2025